AD-A093 850 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/G 5/8
EFFICIENT ENCODING AND DECODING OF SPEECH.(U)
NOV 80 M BEROUTI, M KRASNER, J MAKHOUL MDA903-79-C-0411
UNCLASSIFIED BBN-4567 NL

1 OF 1
AD A093850

END
DATE FILMED
2-81
DTIC

Bolt Beranek and Newman Inc.

LEVEL II

Report No. 4567

Efficient Encoding and Decoding of Speech

Final Report

November 1980

Submitted to:
Mr. David Fonseca, R814
9800 Savage Road
Fort George G. Meade, MD 20755

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Report No. 4567
2. GOVT ACCESSION NO.: AD A09385
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): EFFICIENT ENCODING AND DECODING OF SPEECH
5. TYPE OF REPORT & PERIOD COVERED: Final Report, 1 Nov 79 — 31 Nov 1980
6. PERFORMING ORG. REPORT NUMBER: 4567
7. AUTHOR(s): M. Berouti, J. Makhoul, M. Krasner
8. CONTRACT OR GRANT NUMBER(s): MDA940-79-C-0411
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bolt Beranek and Newman Inc., 10 Moulton St., Cambridge, MA 02238
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: Maryland Procurement Office, 9800 Savage Rd., Ft. George A. Meade, MD 20755
12. REPORT DATE: November 1980
13. NUMBER OF PAGES: 49
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):
15. SECURITY CLASS. (of this report):
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech compression, linear prediction, adaptive predictive coding, down-sampling, quantization, pitch filter, preemphasis, noise shaping, variable-rate output, filter design.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report concludes our work for the past year on efficient coding and decoding of speech. During the past year, we have developed and optimized an adaptive predictive coding (APC) system for high quality speech encoding at 16 kb/s. In our research, we have examined many issues which we described in this report. The system uses the noise-feedback APC configuration, chosen because of the computational efficiencies in implementing spectral noise shaping. The many-level, non-uniform

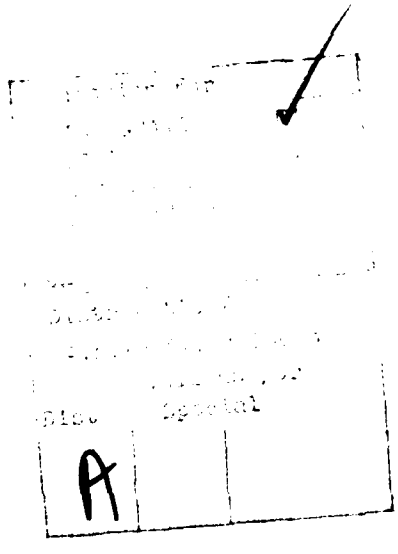
DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

060100

quantizer is derived as a piece-wise linearization of a μ-law non-linear quantizer. The encoding scheme is an entropy coder using a self-synchronizing variable-length code which is matched to the quantization. A simple pole-zero spectral noise shaping is adapted as a function of the input speech spectrum. System parameters are perceptually optimized via listening experiments. As a conclusion to the algorithm development, different implementation strategies were studied as well as methods for decreasing computational complexity. The computational requirements of the algorithm were related to the features of several technologies.



Report No. 4567

EFFICIENT ENCODING AND DECODING OF SPEECH

Final Report
1 November 1979 - 31 November 1980

Prepared by:

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02238

Prepared for:

Mr. David Fonseca, R814
9800 Savage Road
Fort George G. Meade, MD 20755

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In this final report, we present our work performed for the period 1 November 1979 to 30 December 1980 in the area of efficient coding and decoding of speech. Much of the work has been previously reported in project quarterly progress reports and will be summarized. In addition, three topics, time domain noise shaping, direct encoding of the 8 kHz sampled speech signal, and the computational complexity of the algorithm, were investigated during the last contract quarter. These topics are detailed fully. As a conclusion to the report, strategies for efficient algorithm implementation are presented and analyzed based on the results of the project research.

While reading this report, it is important for the reader to keep in mind the goal of this project. The encoding algorithm is designed for the processing of speech already sampled at 8 kHz and quantized at 64 kb/s. The encoded speech is to be stored in digital format at approximately 16 kb/s. The encoding, storage, and decoding processes should not degrade the quality of the speech as measured by preference tests comparing the original and processed speech.

1.1 Organization of the Report

The report is divided into four parts:

- o An overview of the final APC algorithm.
- o The issues of quantization, coding, and stability.
- o Quality improvements through noise shaping and pitch prediction.
- o The computational complexity and implementation of the algorithm.

Section 2 presents an overview of the final APC algorithm. The processing steps are identified and the parameters are given. The details of the optimization of these parameters are discussed in later sections.

Section 3 is mainly concerned with the signal processing and information theoretical aspects of the project. The variable-length coding scheme was compared to optimal coding algorithms and found to be nearly optimal. The adaptive quantization was analyzed and improved. The basic adaptive predictive coding (APC) algorithm was found to be unstable under certain conditions. The cause of this instability was ascertained and a corrective procedure implemented.

Section 4 is primarily concerned with improving the algorithm based on perceptual criteria. Noise shaping, both

temporal and spectral, were implemented and evaluated by listening tests. Inclusion of pitch prediction in the algorithm was also evaluated.

Section 5 is concerned with implementation issues. The computational requirements of the algorithm were examined with the aim of reducing the complexity for a practical implementation. Several prospective architectures are proposed and explored with respect to the existing technology. Issues of performance, flexibility, and cost are examined.

2. OVERVIEW OF THE ADAPTIVE PREDICTIVE CODING (APC) SYSTEM

Adaptive predictive coding (APC) is a simple method of data compression for speech communication. An overview of the compression process as part of a complete input/storage/output system is shown in Fig. 1. In this project, the input speech is digitized before the APC process and has the characteristics as shown: sampled at 8 kHz, bandlimited 300 to 3300 Hz, and corrupted by noise and distortions.

The implementation of the APC system employs the "noise feedback" configuration in Fig. 2. This configuration permits simple (low computational complexity) implementation of the perceptually-optimized spectral noise shaping as discussed in Section 4. The specific parameters used in our implementation are shown in Table 1. In the APC algorithm, the input speech signal is resampled to a 6.67 kHz rate, adequate for the given input speech bandwidth. The processing then proceeds in non-overlapping consecutive frames of 25.5 ms duration. Each frame is windowed by a Hamming window and an optimal eighth-order all-zero inverse filter is computed by a linear prediction recursion. Each filter is quantized via the log-area-ratio (LAR) parameterization to 33 bits. An additional 6 bits per frame is used to quantize a gain parameter. The total bit rate used for parameter specification is 39 bits per frame or 1530 b/s.

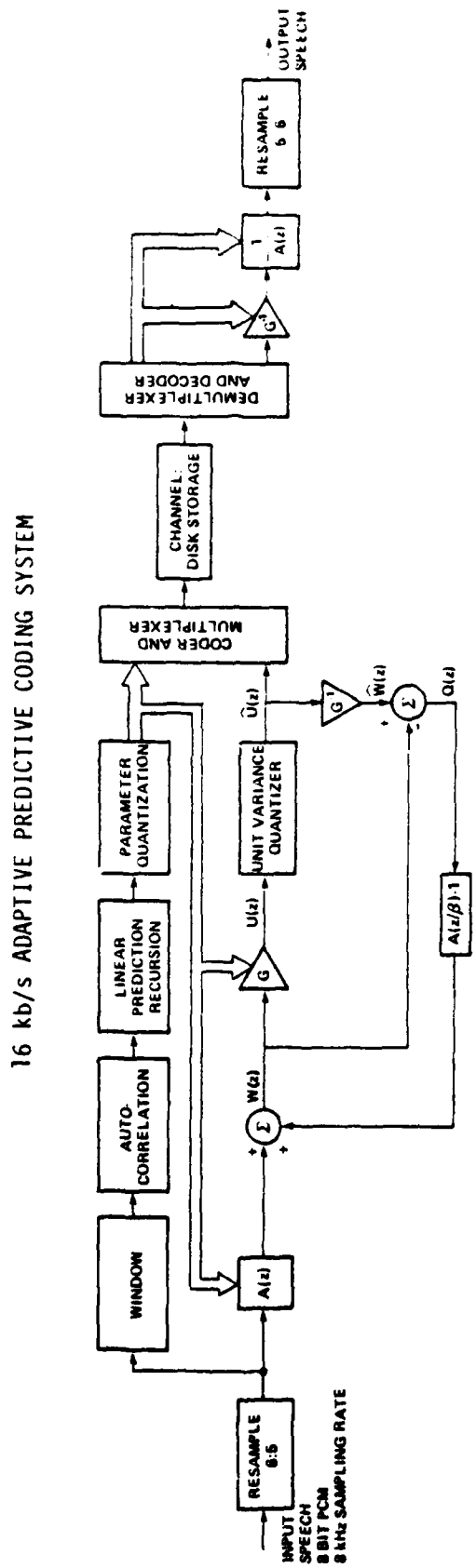
DATA COMPRESSION
OVERVIEW

INPUT SPEECH CHARACTERISTICS:

1. BAND-LIMITED 300-3300 Hz
2. SAMPLED AT 8 kHz
3. CORRUPTED BY:
   A. ACOUSTIC NOISE
   B. QUANTIZATION NOISE (8-BIT PCM)
   C. OTHER DISTORTIONS (E.G., PHASE HITS)
4. $10 \leq SNR \leq 30$ dB

FIG. 1. Overview of the Data Compression System

16 kb/s ADAPTIVE PREDICTIVE CODING SYSTEM



SYSTEM SPECIFICATIONS

- FRAME SIZE : 25.5 MS
  204 SAMPLES AT 8 KHZ
- NON-OVERLAPPING CONSECUTIVE FRAMES
- PARAMETERS : 8 LAR'S : 5 5 5 4 4 4 3 3 = 33 BITS
  1 GAIN : 6 BITS

  TOTAL 39 BITS

  PARAMETER BIT RATE 1530 B/S

- POLE-ZERO NOISE SHAPING:
  DAMPING FACTOR ON POLES : 1.0
  " " " ZEROS : 0.4707 (1600 Hz)
- NOISE FEEDBACK CONFIGURATION
- ADAPTIVE NON-UNIFORM QUANTIZER

  SELF-SYNCHRONIZING CODE: 0,10,110,1110, ...

  AVERAGE = 2.16 $\frac{\text{BITS}}{\text{SAMPLE}}$ 14413 B/S

  TOTAL AVERAGE RATE 16000 B/S

The resampled speech is filtered by the quantized inverse filter, A(z). The output of this filter, the residual, is then input to the APC loop. The design of the adaptive quantizer in the loop, a non-uniform quantizer matched to the variable-length entropy-coding scheme, is discussed in Section 3.4. The quantization error at the system output is colored noise with spectral shape that is a function of the speech spectrum. The pole-zero noise shaping scheme is determined by the APC loop feedback filter, $A(z/\beta)$. This is fully discussed in Section 4.1. An average of 2.16 bits per sample is used in the adaptive quantization. Thus, the total bit rate for quantized parameters and signal is 16 kb/s. The encoded signals are multiplexed for storage on a digital storage medium.

The synthesis is performed by filtering the quantized signal by the all-pole filter, $A^{-1}(z)$. Finally the speech signal is resampled to the original 8 kHz sampling rate.

3. QUANTIZATION, CODING, AND STABILITY

This section considers issues that are related to the design of fundamental processing blocks in the system. The topics are concerned with the signal processing and perceptual aspects of the encoding algorithm. The topics include investigation of fixed-length and variable-length codes, the related quantization schemes, and the stability of the APC feedback loop.

3.1 Variable-Length Coding Techniques

In general, use of a variable-length code allows quantization at a lower distortion (RMS error) for coding at a given bit rate than is possible with a fixed-length code at the same bit rate. This is due to the matching of the lengths of the code words to the sample amplitude probability distribution. Techniques are available that yield bit rates that are arbitrarily close to the entropy of the quantized samples [1].

A subclass of variable-length codes are the self-synchronizing codes. Self-synchronizing codes have the property that the bit stream is uniquely decodable starting in the middle of a codeword sequence. This is a great advantage in a system where transmission errors are possible. The self-synchronizing codes, however, are suboptimal and therefore

may require a larger bit rate than an optimal code for any given quantization. The degree of suboptimality of a particular self-synchronizing code was evaluated for the APC system in order to determine whether the advantages of synchronization outweigh the disadvantages of a larger encoding rate.

An optimal coding scheme requires knowledge of the sample amplitude probability distribution. The actual distribution is calculated and used in determining the code. Since the code varies as a function of the signal, the code (or equivalently the sample amplitude distribution information) must be transmitted to the receiver. The bit rate to transmit this overhead information may be greater than the bit rate savings due to using a better code.

For the APC system, we determined the average bit rate penalty for coding with the self-synchronizing code. The rate for the self-synchronizing code, 2.147 bits per sample, was compared to the entropy of the quantized samples over an utterance and the bit rate using a Huffman code optimized to the sample distribution of the utterance. These rates do not include any overhead information. The average improvement in bit rate in using the Huffman code was 0.003 bits per sample. The entropy was only 0.038 bits per sample less that the self-synchronizing code rate.

From this experiment, we conclude that the penalty for using a self-synchronizing code is minimal. Attempts to optimize the coding scheme may actually increase the bit rate due to the transmission of overhead information.

3.2 Comparison of Variable-Length to Fixed-Length Coding

The majority of APC systems employ fixed-length coding schemes. The advantage of a fixed-length code is that the number of bits per second is a fixed, known quantity. The number of bits that a variable-length code, specified by the average statistics of speech, will use to encode an utterance is a function of the actual statistics of the utterance. The rate may be larger or smaller than the design goal. The problems that this can cause are discussed in Section 3.3.

This section describes listening experiments comparing fixed-length coding schemes to variable-length coding at the same average bit rate. Two types of fixed-length coding schemes were considered. In the first scheme, the gain factor used to adapt the quantizer is updated only once per frame. The quantizer itself is a fixed non-uniform unit-variance quantizer with 4 levels. The quantizer is designed as a function of the statistics of its input for minimum mean-squared error [2]. The quantizer output is encoded with a fixed-length code of 2 bits.

The second scheme employs segmented quantization, either with or without bit-allocation. In segmented quantization, each frame is subdivided into smaller segments, typically up to 10 per frame. One global gain factor is used for the whole frame and, in addition, "delta-gain" values are derived for each segment to account for the difference in energy between a particular segment and the global gain. The quantizer output is again encoded at an average of 2 bits per sample. The delta-gain values are encoded at the rate of approximately 25 bits per frame to maintain a total encoding rate of 16 kb/s.

The delta-gains can be used in two fashions. In the first approach, they are used to normalized each individual segment into a unit-variance signal. In this approach, a 4-level quantizer is used for all samples. A special case of this approach is first scheme described above where there is only one segment per frame. In the second approach, we use the principle of bit-allocation to allocate more or less quantization levels for each segment depending on the value of its delta-gain. This scheme, segmented quantization with bit allocation, requires the availability of several fixed quantizers to be used where appropriate. The possible choices range from 0 bits per sample to 4 bits sample. The optimal choice of how many bits to use for

each segment is given by bit allocation under the constraint that the total number of bits used per frame is constant.

In informal listening tests we compared the outputs of several segmented quantization schemes having 1,3,5 or 10 equal segments per frame, with or without bit allocation, to the output of the entropy-coded variable-rate system, all operating at 16 kb/s.

It was concluded that the entropy-coded system produces a superior output speech quality and, therefore, we continue to use variable-length codes in our final APC system.

3.3 Frame Synchronization using Variable-Length Codes

Channel errors can have a major effect on the performance of the system. In analyzing the effects, we have distinguished between two problems caused by channel errors: sample synchronization and frame synchronization. Sample synchronization is a problem if a channel error causes an erroneous decoding of many samples after the error. The self-synchronizing code eliminates this problem. A channel error will only cause an error in decoding the sample containing the channel error. The error in decoding, however, will be the decoding of two samples when only one was transmitted or decoding

of one sample when two were transmitted. This can cause loss of frame synchronization.

The number of bits used to encode a frame of speech using a variable-length code is not fixed. It can vary depending on the statistics of the speech signal in the frame. The duration of the frame or, equivalently, the number of samples is predetermined. Each frame, information related to the linear prediction filter and the quantization adaptation are multiplexed into the encoded bit stream before transmission. Under the conditions of an error-free channel, it is possible to separate (demultiplex) the parameter data from the coded samples because of the fixed number of samples per frame. If the channel does cause bit errors, the receiver may decode more or less samples than were actually transmitted. Unless specific synchronization information is also transmitted, it may not be possible to determine at the receiver which bits represent frame parameter data and which are encoded samples.

For this reason, we have investigated a scheme that will force the number of bits used to encode each frame to a predetermined constant. Then, the receiver can separate parameter data from coded samples by the number of bits received, not the number of samples. At a frame rate of 40 frames per

second, we allocate 39 bits for parameter data and 361 bits for encoding of the speech residual per frame for a resultant bit rate of 400 bits per frame or 16 kb/s. An additional effect of this conversion to a fixed number of bits in each frame is that no buffering longer than a frame is necessary for transmission over a fixed-capacity synchronous channel.

The method employed to force the number of bits to the required constant is an iterative technique. A frame is quantized and coded by the APC loop. The gain (normalization) of the quantizer is adjusted as a function of the number of bits used for the encoding. This iterative procedure converges rapidly. For a maximum of 5 iterations per frame, the difference between the desired number of bits and the number of bits actually used averages only 7 bits per frame. The algorithm forces the actual number of bits used to be less than the required number. Filler bits are inserted to account for this difference.

This algorithm to fix the number of encoding bits does increase the computation per frame. Each iteration requires the computation of the APC loop including filtering, quantizing, and coding. Since the algorithm will be implemented by the sponsor with a magnetic storage disk as the channel, channel errors are

very infrequent. Therefore, we have eliminated this fixed rate conversion process from the APC algorithm.

3.4 Quantizer Design

The design of the quantizer is dependent on the amplitude distribution of the quantizer input as well as the coding algorithm. The amplitude distribution of the normalized APC residual can be modeled well by a Laplacian probability density. It has been shown that for a Laplacian probability density the optimum mean-squared error quantizer at a given entropy is uniform. Since the variable-length code attains a bit rate nearly equal to the entropy, the uniform quantizer will be close to optimal at a fixed average encoding bit rate using the variable-length code.

We have seen before that a process may be optimal in one particular (and often easy to measure) variable such as mean-squared error but not optimal perceptually. Since the final judge to the quality of the encoded speech is a human listener, we seek to optimize in the perceptual domain. Thus motivated, we experimented with a class of non-uniform quantizers, which, although sub-optimal in a minimum-mean-squared-error sense, could yield results perceptually superior to the case of uniform quantization.

The non-uniform quantization can be implemented as a nonlinearity prior to a uniform quantization with the inverse of the nonlinearity after the uniform quantization. The nonlinearities were fixed, i.e., not adapted as a function of the input signal. Listening experiments were performed comparing the non-uniform quantization to the uniform quantization at the same encoding bit rate.

The listening tests showed that the inverse μ-law with a value of μ=12.5 was perceptually better than the other non-uniform quantizers and the uniform quantizer. The main feature of this non-linearity was that the quantizer step size decreased with increasing amplitude. A simple 2-segment piece-wise linear approximation to the inverse μ-law was then implemented as is shown in Fig. 3. The normalized step size of the middle bin (centered around 0) is 1.13 with the outer bin step sizes of 0.86. The percentages of samples (for our data base) that are quantized into each bin are shown as the vertical hight in the histogram. Also given in Fig. 3 are the variable-length, self-synchronizing codewords assigned to each quantization bin.

Based on the results of listening tests with this simple quantization scheme, we have included this non-uniform quantizer

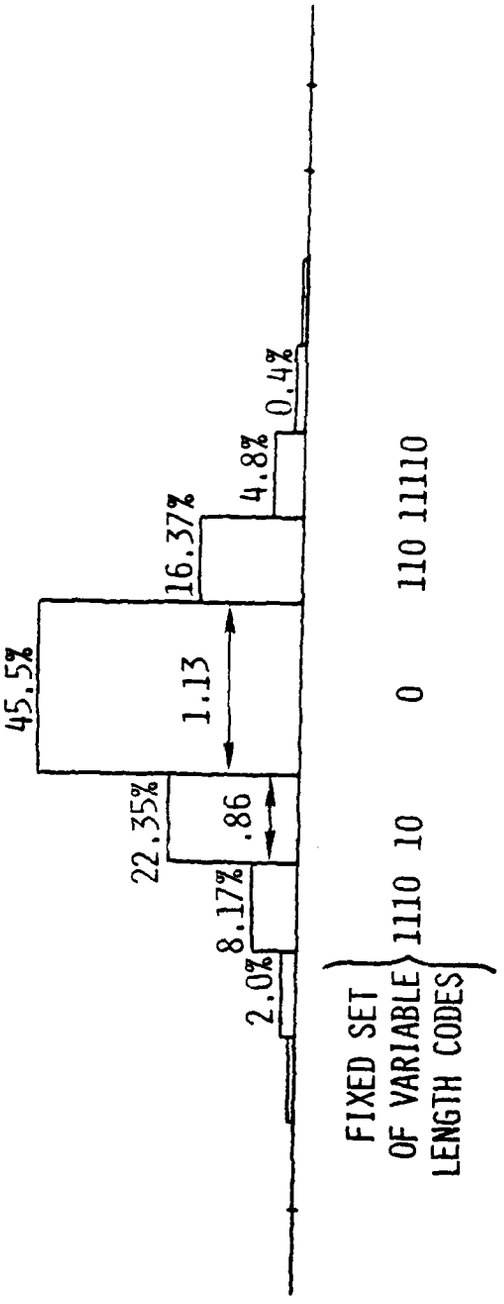


FIG. 3. Non-uniform Quantizer with Variable-Length Codes

into the APC system. This has no effect on the use of the self-synchronizing variable length coding scheme.

3.5 Stability Analysis of the APC System

In our investigation into APC systems, we have noticed that the processed output often contains frames with large amounts of distortion, perceived as "glitches" or "beeps". An examination of the signal to quantization noise (S/Q) for those frames shows a much higher noise level that expected. The usually accurate approximation of the S/Q for the APC system without noise shaping is the product of the linear prediction gain, S/R or V_p^{-1}, and the quantizer input to quantization noise energy ratio, W/Q. Noise shaping reduces the S/Q by some amount (which is a function of the input speech and the particular noise shaping) less than the prediction gain. These high distortion frames can be identified by an S/Q that is much lower than this approximation. Often, the S/Q is negative (in dB), i.e., the noise energy is greater than the speech signal energy.

An analysis of the APC feedback loop, discussed below, indicates that during those frames, the system is not stable. Although the autocorrelation method, used in the linear prediction analysis, guarantees that the all-pole filter is

stable, the stability of the APC feedback loop cannot be guaranteed in general. Because the feedback loop contains a nonlinear element, the quantizer, classical stability analysis techniques cannot be applied directly. By making some reasonable simplifying assumptions, a parametric analysis was performed using the noise-feedback configuration of the system (see Fig. 2).

The power gain (PG) of a filter is the ratio of input to output power for a white noise input signal. When the PG of the APC loop feedback filter, A(z)-1, is greater than the quantizer input to quantization noise power ratio, W/Q, the system is not stable. If a uniform quantizer with variable-length coding is used, the quantization noise level is fixed. Then, W/Q and the bit rate will increase until W/Q is larger than PG. Attempts to iteratively adjust the quantizer to force a fixed bit rate (as described in Section 3.3) will fail because this changes neither PG nor W/Q. There will be no value of quantizer gain-normalization that will yield the required bit rate. If a fixed-length coding scheme were used, the bit rate would always be constant. The quantization noise would increase until the noise can no longer be modeled well by white noise. This has the effect of decreasing the feedback filter PG until the system is stable.

The system is stable if PG is less than W/Q. System performance, however, does suffer if PG is close to W/Q. If W/Q is 5 dB greater than PG, the S/Q is reduced only 2 dB. If W/Q is only 1 dB greater than PG, the S/Q is reduced by 7 dB. Thus, even if the system is stable, performance can be degraded.

There are two schemes we have implemented to solve the stability problem, both relying on reducing the power gain of the feedback filter. In the first method, if the PG is large enough to cause a loss in S/Q of more than 1 dB, the PG is reduced by modifying the feedback filter. This modification is produced by changing the signal autocorrelation vector for the frame and computing a new linear prediction filter. This new filter is no longer the optimal linear prediction filter. The resulting loss in prediction gain, however, is more than compensated for by the increase in S/Q.

The second method is a consequence of the noise spectral shaping scheme we have implemented. The noise spectral shaping algorithm has the effect of modifying the feedback filter in a manner that reduces the power gain. Experimental results show that when the noise spectral shaping described in Section 4.1 is used and there is no iterative modification of the quantizer, then the effect on the bit rate is minimal. For this system

where there is not a requirement for a fixed bit rate (only a fixed average bit rate), noise shaping is the preferred method to reduce the problem of instability. This is discussed additionally in Section 5.1.1

4. NOISE SHAPING AND PITCH PREDICTION

In this section, we consider issues that are directed toward improving the perceived quality of the processed speech. Most of the results presented here are the result of informal listening tests. The results of these experiments show that the value of a parameter that has been optimized perceptually is often different than if the optimization were by signal-to-noise ratio, minimum mean-square error, or other easily measured quantity. The experiments presented here evaluate the effects of spectral and temporal noise shaping and the inclusion of a pitch prediction filter into the system.

4.1 Spectral Noise Shaping

The APC system without spectral noise shaping has an error which can be modeled well by white noise. For a given order of prediction filter, this system is optimal in terms of the minimum mean-square error. We have previously shown that the system with a white noise error is not optimal in terms of perceived quality of the output speech. Spectral noise shaping attempts to improve the quality of the processed speech by minimizing the detectability of the noise. This noise shaping is a dynamic process, adapting at each frame as a function of the input speech signal.

We investigated several different spectral noise shaping schemes. Because of the project emphasis on reduced computational complexity, the noise shaping schemes that we implemented were all simply derived from the all-pole linear prediction estimate of the speech short-time spectrum.

The noise shapings all had poles and/or zeros at the same frequencies as the frequencies of the poles in the linear prediction filter. The bandwidths of the resonances were varied in the experiments by moving the poles and/or zeros closer to the origin in the z-plane. The preferred spectral noise shaping filter is given by

$$B(z) = \frac{A(z/\beta)}{A(z)} \qquad (1)$$

with

$$\beta = 0.4707$$

where $1/A(z)$ is the linear prediction estimate of the speech spectrum and β is the damping parameter. The value of β=0.4707 produces a bandwidth increase of the resonances of 1600 Hz. An example of the spectral noise shaping is shown in Fig. 4. The all-pole model of a typical vowel spectrum is plotted along with the spectral envelopes of quantization noise with and without the noise shaping. Without the noise shaping, the noise is modeled

well by white noise. With the noise shaping, the error is shaped as a function of the all-pole speech spectrum.

This noise shaping, having poles at the same locations in the z-plane as the linear prediction all-pole filter and zeros at the same frequencies but closer to the origin, is simple to implement in the APC noise-feedback configuration. The only additional computation involved is 8 multiplies per 25.5 ms frame to modify the feedback filter coefficients in accordance with the damping. As mentioned in Section 3.5, the modification of the feedback filter to implement the noise shaping lowers the power gain of the filter and reduces the problems due to instabilities of the feedback loop.

4.2 Temporal Noise Shaping

In this section we describe our efforts during the last quarter of this project to control the variations of the output noise energy in APC. Our experiments were motivated by the observation that sometimes the output noise is more audible during some intervals of the speech than in others. It is recalled that we are using an adaptive quantization scheme with a gain factor, at each frame, given by:

$$G_i = 1/\sqrt{E_i} \qquad (2)$$

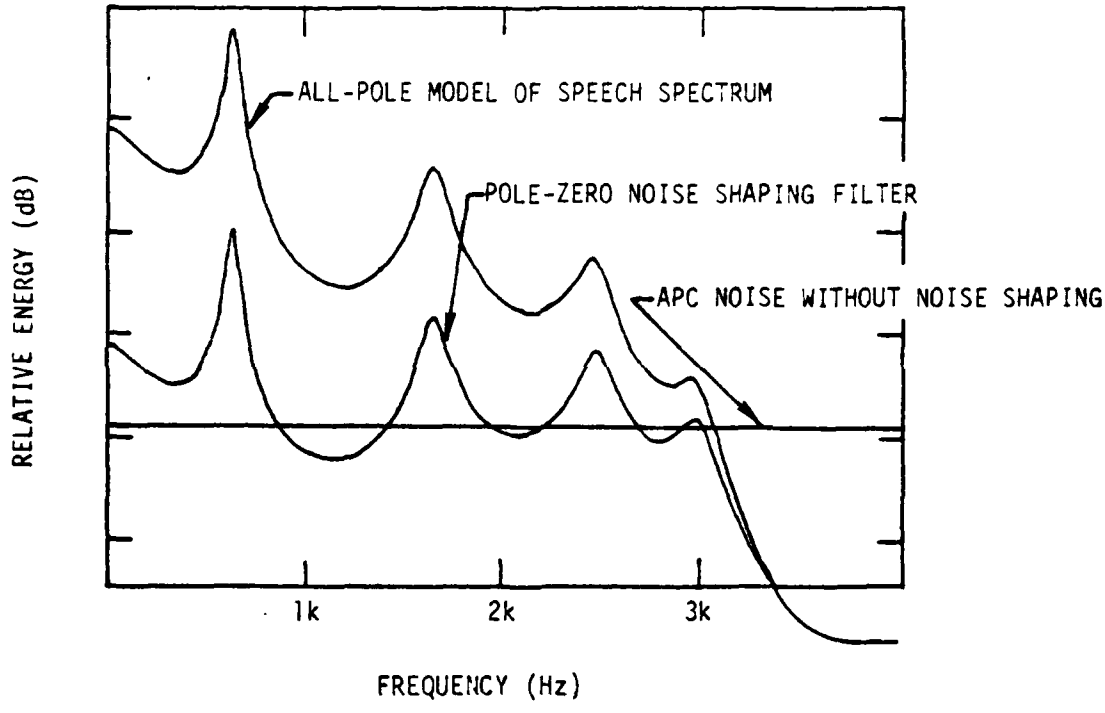


FIG. 4. Spectral Noise Shaping in APC

where E_i is the energy of the linear prediction residual at the i^{th} frame. Thus, the output noise energy in APC is proportional to E_i and varies in time from frame to frame. To have control over the noise level in time we decided to use the following expression for the gain:

$$G_i' = 1/\sqrt{E_i^{\gamma} E_0^{1-\gamma}} \qquad (3)$$

where γ is the new parameter to control the extent of time-domain noise shaping. E_0 is the geometric mean of all E_i values, $1 \leq i \leq M$, where M is on the order of 150 frames. Note that the case $\gamma=1$ is a special case of noise shaping that results automatically from the conventional implementation of APC. For $\gamma=0$ there is no noise shaping because the gain G_i' is constant and equal to $1/\sqrt{E_0}$. For that case, the output noise level is constant in time and proportional to E_0.

Under the condition that E_0 be the geometric mean value of E_i, it can be shown that all cases operate at the same average bit-rate and at the same average segmental signal-to-noise ratio (SNR). What is different for different values of γ is the total output noise power, measured over M frames, and the manner in which the noise energy varies in time. The effect of non-zero values of γ on the operation of the APC system is the trading of bits among frames, such that the SNR of some frames is improved

at the expense of decreasing the SNR of other frames, while maintaining the same average bit-rate.

We experimented with different values of γ. For $\gamma<1$, the SNR improves at those frames where $E_i>E_0$ relative to the case with $\gamma=1$. However, the SNR decreases at those frames where $E_i<E_0$, relative to the case with $\gamma=1$, and the SNR decreases where $E_i>E_0$.

For a clean speech data-base, informal listening tests showed that a value $\gamma=0.9$ is perceptually optimal. However, for the noisy speech data-base used in this project we were not able to find a value of γ that yields results perceptually superior to the case $\gamma=1$. For that reason, we have not included time-domain noise shaping in the final APC system.

4.3 Pitch Prediction

The prediction filter determined by the eighth-order linear prediction algorithm is a "short-time" prediction based on the first eight terms of the speech autocorrelation vector. Speech, however, has a large correlation at delays equal to the pitch period, on the order of 3 to 20 ms. The pitch predictor is a second predictor filter to account for this "long-time" correlation.

For this investigation, we implemented pitch predictors of order n, $1 \leq n \leq 5$. The pitch predictor is implemented as an n-tap finite impulse response (FIR) filter. The pitch predictors were implemented in the APC system with spectral noise shaping.

Listening experiments using noise-corrupted speech utterances were performed comparing the system with each of the 5 pitch predictors to the system without pitch prediction. Although there was a slight improvement in quality with the pitch prediction, we feel that the improvement was not adequate to offset the additional computational cost of the implementation.

5. ALGORITHM IMPLEMENTATION

In this section, we consider the issues germane to efficient implementation of the APC speech coding algorithm. The algorithm resulting from our research described in this report is examined to determine its computational complexity. To reduce the amount of computation required for the algorithm, several modifications have been examined. These modifications and their effect on system performance are described. Finally, we discuss several possible architectures for implementation of the system.

5.1 Computational Complexity of the APC Algorithm

The APC system is comprised of two major tasks: analysis and synthesis. The implementation of these tasks may be of very different architectures. This reflects differences in computational complexity and in the requirements of input and output performance. In Fig. 5, the system flowchart is annotated with the approximate number of thousand multiply-accumulate (KMAC) operations per second. This is a reasonable approximation of the complexity of the algorithm for a processor where computational speed is the overall limiting factor. Since the APC system requires much filtering, this assumption should be true for most processor implementations. We see from the

flowchart that over half of the total computation is to implement the resampling operations. This is discussed in Section 5.1.2 and 5.1.3.

5.1.1 APC Feedback Loop Stability

In Section 3.5, the issue of the stability of the APC feedback loop was discussed. Two methods of solution for the stability problem were proposed. Both methods rely on reducing the power gain (PG) of the feedback filter. The first method uses an iterative technique to modify the prediction filter found in the linear prediction analysis. By adjusting the autocorrelation vector of the input speech frame, a filter with smaller power gain is produced with little loss in prediction gain. Referring to Fig. 5, the linear prediction recursion must be performed for each iteration. The windowing of the frame and the autocorrelation calculations are not repeated. From the standpoint of computational requirements, this technique is not very costly. The other method, however, requires no additional computation.

The second method uses the spectral noise shaping in the noise-feedback APC configuration. The modification of the feedback filter to implement the desired noise shaping has the effect of reducing its power gain. This method, therefore, is preferred for use in the system.

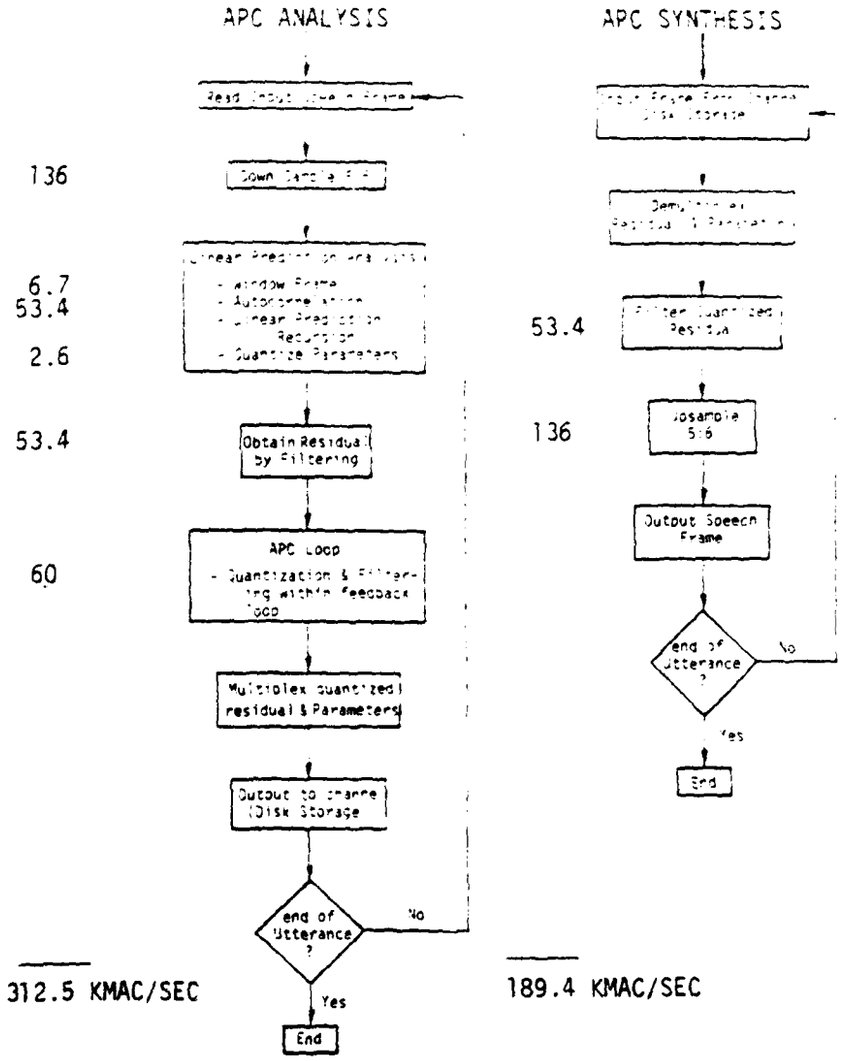


FIG. 5. Computational Requirements of APC Analysis and Synthesis

5.1.2 Efficiencies in Resampling

As seen in Fig. 5, the resampling operations account for over half of the computation involved in the APC system. The resampling is implemented by interpolation and decimation operations involving filtering. Because of the decimation involved, finite impulse response (FIR) filters do not require more computation than would infinite impulse response (IIR) filters. A shorter length FIR filter, however, would reduce the computation.

The original algorithm used an equal-ripple design FIR filter of length 250. The results of our listening tests show that a Hanning window design FIR filter of length 100 is adequate. This results in a savings of 60% of the computations over the system with a filter of length 250. This assumes that the system is used with 3.8 kHz lowpass anti-aliasing filters before the A/D converter and after the D/A converter. This filter of length 100 is used in determining the number of calculations in Fig. 5.

Further listening tests were performed with filters of length 64. This would yield an additional 36% savings in computation over the FIR filters of length 100. These filters degraded the processing for the system with 3.8 kHz lowpass

anti-aliasing filters. When 3.2 kHz lowpass filters were used, the degradation was not audible. Thus, the sponsor should consider using 3.2 kHz lowpass anti-aliasing filters in the synthesis stage of the system.

5.1.3 Coding of the Residual Sampled at 8 kHz

The input to the APC system is noise-corrupted speech that has been bandlimited to 300 - 3300 Hz. The sampling of the speech signal is at 8 kHz, adequate to represent a 4 kHz bandwidth signal. Since the system encoding bit rate is fixed at 16 kb/s, the average number of bits per sample may be increased by sampling at a frequency of less than 8 kHz. With this motivation, the resampling operations have been included in the encoding algorithm. The input speech is resampled from 8 kHz to 6.67 kHz before processing and resampled from 6.67 kHz back to 8 kHz at the output. This change of sampling rate increases the average number of bits per sample from 1.789 to 2.147. The effect is to increase the average residual-to-noise ratio (W/Q) from 9.2 dB to 11.0 dB, an increase of 1.8 dB. If we consider only the noise that is within the speech band, the increase is only 1.0 dB.

Because the resampling operations require a large amount of algorithm computation, we investigated methods of eliminating

resampling from the algorithm. The resulting computational requirements due to coding directly at 8 kHz are shown in Fig. 6. The computation for analysis decreased by about 32% while the computation for synthesis drops by 66%. The number of operations assumes that 8 pole linear prediction analysis is still sufficient. Using 10 pole analysis would reduce the savings due to 8 kHz coding.

It was conjectured that our improvements to the algorithm quality would allow the slight degradation caused by having a lower W/Q. Since this was a major modification to the algorithm, several of the parameters, especially the spectral noise shaping, needed to be reoptimized for the direct encoding at the 8 kHz sampling rate.

First, we investigated a technique designed to take advantage of the oversampling of the signal at 8 kHz. This technique was designed to reduce the noise within the speech band while increasing the noise in the 3.3 to 4.0 kHz region by a spectral noise shaping scheme. Examination showed that the maximum gain in S/Q to be realized from this technique was less than 1 dB. We decided not to use this technique because of the additional computation required to achieve this gain.

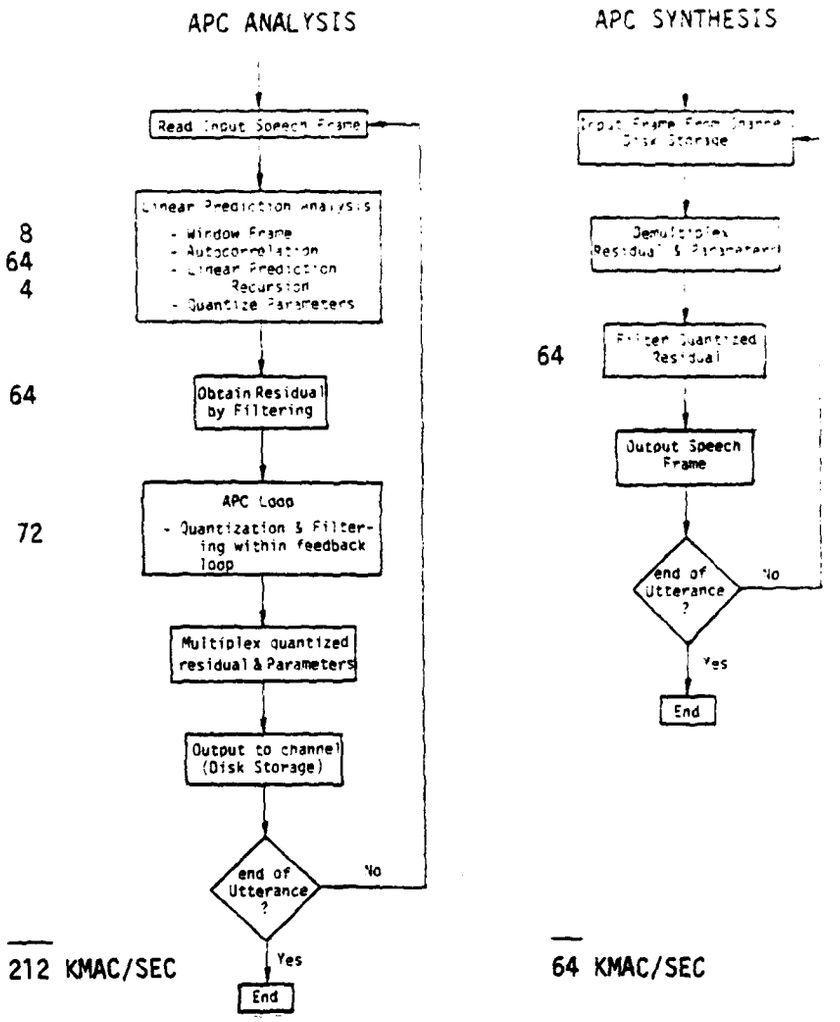


FIG. 6. Computational Requirements of APC Analysis and Synthesis Without the Resampling Operations

Next, we investigated direct encoding at the 8 kHz rate with no additional processing. Reoptimization of the spectral noise shaping damping factor resulted in a value of 0.5682, equivalent to a 1200 Hz bandwidth increase of the resonances. Listening experiments showed that trained listeners could distinguish between the original and processed utterances. The degradation of the processing was slight and may not result in any differences between original and processed utterances in a preference test.

A final experiment included a 3.2 kHz lowpass anti-aliasing filter at the output of the D/A converter. The use of this filter reduced the audibility of differences caused by the system processing. If the sponsor were to use a 3.2 kHz filter in the system, we believe that the resampling operations could be eliminated.

5.2 Architectures for Implementation

There are several possible implementation strategies for the APC system. A brief review of the system performance requirements will help to clarify the discussion regarding the features of each proposed architecture. Several technologies are reviewed as to their performance features. This information is

then used in a discussion of the applicability of each of the technologies for implementation of the APC system.

5.2.1 Input and Output Channel Requirements

The input speech for the APC analysis has been digitized at an 8 kHz sampling rate at 64 kb/s. Three possible sources have been identified:

1. The speech may be stored on a random access magnetic storage device. The retrieval and compression of the speech events should keep up with the rate of new speech being entered into the storage.
2. These speech events may be put on a 1.024 Mb/s digital data stream, representing a single voice channel that has been speeded up by a factor of 16, i.e., the speech input channel is 16 times real-time. The analysis processor should be able to perform at 16 times real-time.
3. The speech events may be contained on a 1.544 Mb/s communication line. This corresponds to a maximum speech rate of approximately 24 times real-time.

To keep up with the input speech events, the analysis processor must have an average processing rate greater than the maximum rate of speech on the source channel.

The synthesized speech output must be available to 64 or more independent listening stations at all times. Each of the listening stations must be able to quickly access and play any speech utterance stored on the system. This may be performed by directly communicating with the listening stations or by buffering through a storage medium.

5.2.2 Present Technology Performance

There are several technologies that offer the performance necessary for real-time operation. The basic requirement is for computational power. Both analysis and synthesis require a processor designed for fast multiply-accumulate operations. We investigated the performance specifications of three technologies that are appropriate for the APC system implementation:

1. Array Processor Implementation.
2. μ-Processor & Signal Processor Chip Implementation.
3. Custom Very Large Scale Integration (VLSI) Implementation.

Array processors have several important advantages over other approaches. When purchased, they are complete and (hopefully) debugged processor systems ready for interfacing to the main system. Their ability to be programmed, sometimes in higher level languages, results in relatively low development costs. Hardware costs per processing unit, however, are much higher than the other possible technologies.

Two examples of array processors are the Floating Point Systems AP120B and the Culler-Harrison Systems CHI-5. The AP120B provides floating-point computation with a 167 ns cycle time. It has been field-proven to be reliable. Price per unit ranges from

$50K to $100K depending on options. The CHI-5 has integer and floating-point computational modes with a basic cycle time of 250 ns. Price is estimated at under $10K, but no units have been delivered as of this date. Culler-Harrison Systems, in conjunction with Motorola, has already begun development of a smaller, cheaper, and less power consuming model of the CHI-5 using a VLSI aritnmetic processing chip.

Another possible implementation is to develop a processing module based on a signal processing chip with µ-processor for logical control. While development is more costly than an already designed array processor system, production costs per unit is significantly lower as are the size and power consumption. Several appropriate signal processor chips are presently on the market. These include products by Nippon Electric Corp., American Microsystems, Inc., and TRW. It is expected that several other companies will introduce similar signal processing chips in the near future.

All of the presently available signal processor chips perform fixed-point computation. Multiply-accumulate operation accuracy of these different chips range from a 12 bit by 12 bit multiply with 16 bit accumulate to a 16 bit by 16 bit multiply with a 32 bit accumulate. The truncation (quantization) effects

in computation can result in degradation of speech quality if the word lengths are not sufficient. Cycle times for these processors range from 140 to 300 ns.

A third possible technology is that of custom VLSI chips. Present VLSI technology allows implementation of many small systems on single chips, e.g., speech synthesis chips and μ-processors. It is estimated that the complexity possible on a chip will increase by several orders of magnitude in the next decade. Much larger systems can then be fabricated on a single chip. VLSI is the smallest of the possible implementations. Although development costs are high, each delivered unit is very inexpensive. Power requirements are also minimal. Also, computer-aided design techniques for the automated design of VLSI systems will reduce VLSI development costs significantly in the next several years.

5.2.3 Possible Architectures for Analysis and Synthesis

As we have seen in Section 5.1, the analysis process requires more computation and logical control than does the synthesis process. At 16 or more times real-time (and assuming no resampling operations), the analysis would require 3650 thousand multiply-accumulate operations (KMAC) per second or one MAC per 275 ns. An array processor, the AP120B can perform a

floating point MAC every 167 ns. Several signal processing chips and/or custom VLSI can also perform multiply-accumulate calculations in less than 275 ns.

The synthesis process is much simpler than analysis. One synthesizer requires only 64 KMAC per second or one MAC per 15 μs. For 64 listening stations, 64 times real-time operation would be required if all stations were in use at once. This would require 4096 KMAC per second or one per 244 ns.

It should be noted that the problems of implementation for the 64 times real-time synthesis is very different than those encountered in implementing the 16 times real-time analysis. The data source for the APC analysis is one serial data channel with each speech event being a contiguous data stream. Thus, a single processor or a pipelined multi-processor architecture is a natural choice for the analysis. The output for the synthesis is 64 simultaneous, independent speech events. While it is possible to have a single fast processor performing synthesis for all listening stations at once, logical control for the multiplexing of processing between the 64 speech events may be a problem. An obvious solution is a parallel multi-processor approach using a single processor for each of the listening stations. Each of these processors has relatively little computation to perform.

Control of communication to the listening stations is simple. A possible disadvantage due to a parallel multi-processor approach is the need to allow each processor access to the speech data on the main magnetic storage.

In choosing between technologies for the analysis and synthesis operations, it is important to consider the contribution of development costs and hardware costs to the total cost. For a single processor per system, hardware cost may be a small part of the total cost. A more expensive array processor could be very cost effective. Alternatively, if 64 or more processors were needed to perform an operation, the hardware cost may be the major part of the total cost. Custom VLSI might then be the most cost effective approach.

We have investigated some of the issues regarding architectures for implementation of the APC speech system. Our calculations of computational complexity show that due to the requirements of many times real-time operation for both analysis and synthesis, some of the possible implementations would be near the present limitations of the chosen technology. We feel that an in-depth study including preliminary designs of the processors in each of several technologies is necessary before final design and fabrication/construction.

REFERENCES

1. Gallegher, Information Theory, John Wiley & Sons, Inc., 1968.

2. Max, "Quantizing for Minimum Distortion," IRE Trans. on Info. Theory, 1960, pp. 7-12.

END

DATE
FILMED

2-81

DTIC